

? s pn=63289646  
S15 1 S PN=63289646

? t s15/full

15/19/1 Links

JAPIO

(c) 2005 JPO & JAPIO. All rights reserved.

02672746 \*\*Image available\*\*

# PROGRAM MODULE MANAGING SYSTEM

**Pub. No.:** 63-289646 [JP 63289646 A ]

**Published:** November 28, 1988 (19881128)

**Inventor:** OKAMOTO KEISUKE

**Applicant:** NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)

**Application No.:** 62-125059 [JP 87125059]

**Filed:** May 22, 1987 (19870522)

**International Class:** [ 4 ] G06F-009/44; G06F-009/06

**JAPIO Class:** 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)

**Journal:** Section: P, Section No. 845, Vol. 13, No. 116, Pg. 31, March 22, 1989 (19890322)

## ABSTRACT

**PURPOSE:** To highly efficiently execute processing corresponding to debugging by allowing a table forming means to form a control table in accordance with an input referring to a display menu and controlling an external storage device and a main storage device in accordance with addresses in the table.

**CONSTITUTION:** A menu is displayed on a display device 2 by a menu control part 3 in a processor 16 in accordance with start based upon an input device 1. In accordance with the operation of the device 1 based upon the display, the table forming part 4 forms a source module table, an object module table and a load module table corresponding to a specified program by successively referring to a file table, a module table and respective program module tables in a file control table 14 stored in a storage part 17. A program module corresponding to an address shown in these tables is stored in a hard disk device in the external storage device and read out from the device 12 to the main storage device. Consequently, a necessary address can be determined without executing complex and easily missed calculation and the debugging processing can be highly efficiently executed.

## ⑫ 公開特許公報(A)

昭63-289646

⑤ Int. Cl.

G 06 F 9/44  
9/06

識別記号

3 2 2  
3 1 0

庁内整理番号

B-8724-5B  
B-7361-5B

⑬ 公開 昭和63年(1988)11月28日

審査請求 未請求 発明の数 1 (全6頁)

⑭ 発明の名称 プログラムモジュール管理方式

⑮ 特 願 昭62-125059

⑯ 出 願 昭62(1987)5月22日

⑰ 発 明 者 岡 本 恵 輔 東京都港区芝5丁目33番1号 日本電気株式会社内

⑱ 出 願 人 日本電気株式会社 東京都港区芝5丁目33番1号

⑲ 代 理 人 弁理士 境 廣 巳

## 明 細 書

## 1. 発明の名称

プログラムモジュール管理方式

## 2. 特許請求の範囲

ソースモジュール、オブジェクトモジュール、ロードモジュールの各プログラムモジュールを外部記憶装置に格納する際の位置情報、前記外部記憶装置に格納された各プログラムモジュールを読出す主記憶装置の位置情報を指示する制御テーブルと、

該制御テーブルを作成するテーブル生成手段と、  
該テーブル生成手段による前記制御テーブルの作成時に使用される処理メニューを入出力装置に表示し、前記処理メニューに従って前記入出力装置から入力される情報を前記テーブル生成手段に渡して前記制御テーブルを作成させるメニュー制御手段とを備え、

ソースモジュール、オブジェクトモジュール、ロードモジュールを前記外部記憶装置に格納するとき、および前記主記憶装置に読出すときには前

記制御テーブルに設定された位置情報に従わせることを特徴とするプログラムモジュール管理方式。

## 3. 発明の詳細な説明

(産業上の利用分野)

本発明は、ソースモジュールなどのプログラムモジュールを外部記憶装置に格納し、或いは主記憶装置に読出す際の格納位置を指示する制御テーブルをキーボード等の入力装置からの入力で作成し、その制御テーブルの内容に従ってプログラムモジュールの外部記憶装置および主記憶装置の格納を制御するプログラムモジュール管理方式に関する。

(従来技術)

ソースモジュール、これをコンパイルしたオブジェクトモジュール、複数のオブジェクトモジュールをリンクしたロードモジュールなどのプログラムモジュールを計算機システムの外部記憶装置に格納し、その後何等かの処理を行なうために主記憶装置に読出す場合、従来は、JCL(ジョブ制御言語)を使用して外部記憶装置への格納や主

記憶装置への読出し等の処理の実行を指示している。

(発明が解決しようとする問題点)

このような方法では、JCLに関する理解が必要であり、またJCLはカードリーダから入力されるのでカードリーダが備えられているソフトウェアセンタ等の場所に向かなければならないという問題点がある。

また、各プログラムモジュールを外部記憶装置のどの場所に格納するのか及び外部記憶装置から読出したプログラムモジュールを主記憶装置のどの場所に格納するのかは、従来、オペレーティングシステムメモリ割付機能により自動的に行なわれており、それらのメモリ割付場所を簡単に知ることができなかったことから、デバッグ時に記憶装置のリード、ライトを行なうときアドレス算出が面倒となり、デバッグが効率良く行なえないと共にアドレス算出ミスによりプログラムの修正誤りも起こり易いという問題点もある。

本発明はこのような従来の問題点を解決したも

のであり、その目的は、プログラムモジュールを格納する外部記憶装置および主記憶装置の場所を指示する格納情報を持つ制御テーブルをキーボードなどの入力装置からの操作で作成することができ、各プログラムモジュールの外部記憶装置および主記憶装置上での格納位置を指示可能としたプログラムモジュール管理方式を提供することにある。

(問題点を解決するための手段)

本発明は上記目的を達成するために、ソースモジュール、オブジェクトモジュール、ロードモジュールの各プログラムモジュールを外部記憶装置に格納する際の位置情報、前記外部記憶装置に格納された各プログラムモジュールを読出す主記憶装置の位置情報を指示する制御テーブルと、

該制御テーブルを作成するテーブル生成手段と、

該テーブル生成手段による前記制御テーブルの作成時に使用される処理メニューを入出力装置に表示し、前記処理メニューに従って前記入出力装置から入力される情報を前記テーブル生成手段に

渡して前記制御テーブルを作成させるメニュー制御手段とを備え、

ソースモジュール、オブジェクトモジュール、ロードモジュールを前記外部記憶装置に格納するとき、および前記主記憶装置に読出すときには前記制御テーブルに設定された位置情報に従わせる構成を有する。

(作用)

各プログラムモジュールの外部記憶装置および主記憶装置上での格納位置を指示する制御テーブルがキーボードなどの入力装置からの操作によって作成できるので、従来のようなJCLを使用する必要がなくなり、また、プログラムモジュールの格納位置を任意に設定できるので、デバッグ時の効率を高めることができる。

(実施例)

次に本発明の実施例について図面を参照して説明する。

第1図は本発明の実施例のブロック図であり、キーボードなどの入力装置1と、CRTなどの表

示装置2と、処理装置16と、フロッピーディスク装置などの補助記憶装置5と、プログラムファイル15を格納するハードディスク装置12とで構成されている。また処理装置16には、テーブル処理メニュー画面の表示入力制御を行なうメニュー制御部3と、テーブル生成部4と、プログラム読込部6と、コンパイラ7と、リンカ8と、ロード等の転送部9と、主記憶装置10と、実行部11と、記憶部17とを含む。この記憶部17は主記憶の一部を構成し、サブコマンドテーブル13、ファイル制御テーブル14が格納される。ハードディスク装置12のプログラムファイル15にはファイル制御テーブル14の内容に従って、ソースモジュール(以下SMと称す)、オブジェクトモジュール(以下OMと称す)、ロードモジュール(以下LMと称す)の各プログラムモジュールが格納される。

第2図はファイル制御テーブル14の構成例を示す。このような構成のファイル制御テーブル14を本実施例では入力装置1からの入力により生成するものである。同図に示すファイル制御テーブル

14は、ファイルテーブル140、モジュールテーブル141、141<sub>1</sub>、SMテーブル143、OMテーブル144、LMテーブル145とで構成される。ファイルテーブル140の各エントリにはファイル名(ABCD等)とモジュールテーブル141<sub>1</sub>、141<sub>2</sub>へのポインタ(MP<sub>1</sub>等)とが格納され、そのポインタによって各モジュールテーブル141<sub>1</sub>、141<sub>2</sub>がファイルテーブル140にチューンされる。また、各モジュールテーブルにはモジュールテーブル141<sub>1</sub>に例示するように、SMテーブル143へのポインタ(SMP<sub>1</sub>等)、OMテーブル144へのポインタ(OMP<sub>1</sub>等)、LMテーブル145へのポインタ(LMP<sub>1</sub>等)が格納され、モジュールテーブル141<sub>1</sub>にチューンされている。SMテーブル143、OMテーブル144、LMテーブル145には、更新日、更新回数、予約サイズ、現在サイズ、ハードディスク装置の格納先頭アドレス(HDアドレス)、ローディング時の主記憶先頭アドレス(MMアドレス)等が格納される。

次にファイル制御テーブル14の作成処理について

とモジュール識別情報とをテーブル生成部4へ渡す(ステップS13)。テーブル生成部4はこれに回答して第2図のファイルテーブル140の指定されたエントリにファイル名ABCDを登録し、且つ、対応するモジュールテーブル141<sub>1</sub>とSMテーブル143の枠組みが作成されていなければそれらを作成し、ポインタMP<sub>1</sub>、ポインタSMP<sub>1</sub>をファイルテーブル140、モジュールテーブル141<sub>1</sub>の対応する部分に格納する。

次に入力装置1からリターンのメッセージが受信されると、ステップS12の処理で判別され、第3図のステップS1へ戻り、表示装置2の画面に再び第4図に示すテーブル処理メニューが表示される。

次に入力装置1よりメニュー番号2のファイル制御テーブル表示・更新を選択するメッセージが受信されると(ステップS2)、メニュー制御部3はメニュー番号2の処理を行なう(ステップS4)。

第6図はメニュー番号2の処理を行なうステッ

で説明する。

操作者が入力装置1から起動信号を入力すると、メニュー制御部3が起動され、メニュー制御部3は例えば第3図に示す処理を開始し、例えば第4図に示すようなテーブル処理メニューを表示装置2の画面に表示する(ステップS1)。

次に入力装置1よりメニュー番号1のファイル名更新を選択するメッセージが受信されると(ステップS2)、メニュー制御部3はメニュー番号1の処理へ移行する(ステップS3)。

第5図はメニュー番号1の処理を行なうステップS3の処理例であり、先ず、記憶部17のファイル制御テーブル14における第2図に示したファイルテーブル140を読み込んでこれを表示装置2の画面に表示する(ステップS10)。

次に入力装置1からファイル名として例えばABCDを示すメッセージとモジュール識別情報としてSMを示すメッセージとが受信されると(ステップS11)、ステップS12を経てメニュー制御部3は受信したメッセージつまりファイル名情報

とモジュール識別名のメッセージ受信待ちとなる(ステップS20)。入力装置1より例えばファイル名ABCD、モジュール識別名SMのメッセージが与えられると、メニュー制御部3はこれを識別し、ステップS21、S22を経てステップS23へ進み、記憶部17中のファイル制御テーブル14における第2図のSMテーブル143を例えば第7図に示す形式で表示装置2の画面に表示する(ステップS23)。

操作者は第7図に示された内容の表示画面を見ながら、後で入力するSMの予約サイズ、格納先となるハードディスク装置12上のアドレス(HDアドレス)と、ローディング時の主記憶装置10上のアドレス(MMアドレス)とを入力する。この入力されたHDアドレスなどは第6図のステップS20で受信され、メニュー制御部3はステップS21、S22を経てステップS24によりHDアドレスなどの入力情報をテーブル生成部4へ渡す。テーブル生成部4はこの入力情報を記憶部17のファイ

ル制御テーブル14に書込む。なお、更新日、更新回数、現在サイズは自動的に設定される。操作者が必要な情報を設定し終わるとリターンのメッセージを入力装置1から入力することにより、処理を終える。

さて、以上のようにして第2図のSMテーブル143の作成を行なった後、例えば入力装置1からファイル名ABCD、モジュール種類がSMであることを指示して、作成したSMの各ステートメントを入力装置1或いは補助記憶装置5から入力すると、これらはプログラム読込部6に与えられる。そして、プログラム読込部6はファイル制御テーブル14のSMテーブル143のHDアドレスを参照し、ハードディスク装置12の対応部分へ入力されたSMを記憶し、SMテーブル143の現在サイズを設定する。これによって、ハードディスク装置12の記憶域のうち操作者が事前に指定されたアドレス部分にSMを格納することが可能となる。

操作者はSMテーブル143を作成したと同様の操作を行なうことにより、第2図のOMテーブル

144を作成し、そのOMテーブル144のHDアドレス、MMアドレスに所望の値を設定した後、入力装置1からコンパイラ7にファイル名ABCDのSMをコンパイルすべき指示を与えると、コンパイラ7はSMテーブル143のHDアドレス、現在サイズを参照してSMを読み出し、これを翻訳してOMテーブル144に設定されたHDアドレスに従ってハードディスク装置12の領域にOMを格納し、また現在サイズを設定する。

更に操作者が、HDアドレス、MMアドレスに所望の値を設定したLMテーブル145を作成し、リンクすべき複数のOMを指定してリンク8に指示を与えると、リンク8は対応するOMテーブルを参照してリンクすべきOMをハードディスク装置12から読取ってリンク処理を行ない、これを指定されたLMテーブル145に設定されたHDアドレスに従ってハードディスク装置12の領域に格納し、現在サイズを設定する。

次に、デバッグを行なうために、ファイル名ABCDのLMを実行させる場合、操作者が入力装

置1からファイル名ABCD、モジュールの種類LMを指定すると、転送部9が起動され、転送部9はファイル制御テーブル14からLMテーブル145を参照し、そのHDアドレスで指示されたハードディスク装置12の領域からLMを読み込んでそのLMテーブル145のMMアドレスの位置から順に主記憶装置10へロードしていく。そしてロード完了後、そのLMは実行部11によって実行される。

また本実施例では、表示装置2の画面にLMテーブル145の内容を表示させることにより、操作者は実行中のLMの先頭アドレス(MMアドレス)を簡単に知ることができるので、LMの任意のアドレスを指定したLMの実行が複雑なアドレス計算無しに容易に行なうことができ、デバッグの効率を高めることができ、LMのバッチによって修正すべき箇所のアドレスも簡単に知ることができる。更に、転送部9にSM、OMの主記憶装置10への読み込みを入力装置1から指示した場合、SMテーブル143、OMテーブル144に設定されたMMアドレスに従った位置に読み込まれるので、SM、

OMのデバッグも容易に行なうことができる。

なお、メニュー制御部3は、第4図のメニュー番号3のファイル一覧表表示が選択されれば、ファイルテーブル140、各モジュールテーブル141<sub>1</sub>、141<sub>2</sub>の内容を表示装置2の画面に表示する。また、メニュー番号4のサブコマンドテーブル13が選択されれば、記憶部17に格納されたサブコマンドテーブル13の内容を表示装置2の画面に表示する。ここで、サブコマンドテーブル13は、リンク8の起動時等に使用するサブコマンドを格納する為のものであり、操作者はそのサブコマンドを使用してリンク8等を与えるコマンドを選択することができる。

(発明の効果)

以上説明したように、本発明によれば、メニュー制御手段によってCRT等の出力装置に表示されるテーブル処理メニュー画面に従ってキーボード等の入力装置から必要なデータを入力していくことで、プログラムモジュールの外部記憶装置および主記憶装置上での格納位置を指示するSMテ

ーブル等の制御テーブルを、JCLを使用することなく作成することができるので、オペレーティングシステムの専門知識がない操作者も容易にプログラムモジュールの管理を行なうことが可能となり、且つ、SM、OM、LMを外部記憶装置に格納するとき、および主記憶装置に読出すときには前記制御テーブルに設定された位置情報に従わせるものであるから、プログラムモジュールの格納場所を操作者が制御テーブルに設定した場所通りとすることができ、デバッグ時の効率を高めることが可能となる。

#### 4. 図面の簡単な説明

第1図は本発明の実施例のブロック図、

第2図はファイル制御テーブル14の構成例を示す図、

第3図はメニュー制御部3の処理例の流れ図、

第4図はテーブル処理メニュー画面の内容例を示す図、

第5図はメニュー番号1の処理例の流れ図、

第6図はメニュー番号2の処理例の流れ図および、

び、

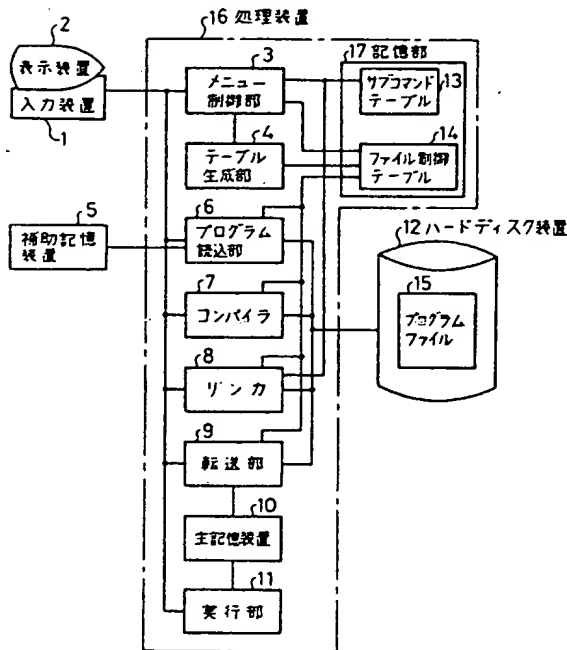
第7図はSMテーブル表示画面の内容例を示す図である。

図において、

- |            |               |
|------------|---------------|
| 1…入力装置     | 10…主記憶装置      |
| 2…表示装置     | 11…実行部        |
| 3…メニュー制御部  | 12…ハードディスク装置  |
| 4…テーブル生成部  | 13…サブコマンドテーブル |
| 5…補助記憶装置   | 14…ファイル制御テーブル |
| 6…プログラム読込部 | 15…プログラムファイル  |
| 7…コンパイラ    | 16…処理装置       |
| 8…リンカ      | 17…記憶部        |
| 9…転送部      |               |

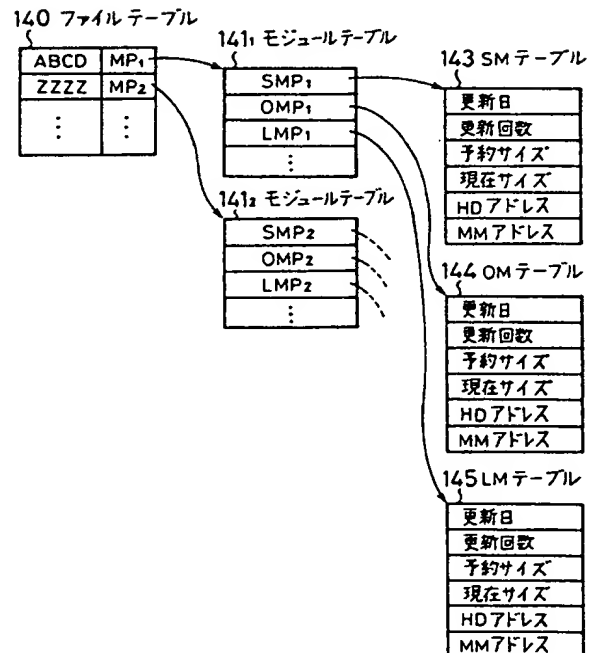
特許出願人 日本電気株式会社

代理人 弁理士 境 廣 巳



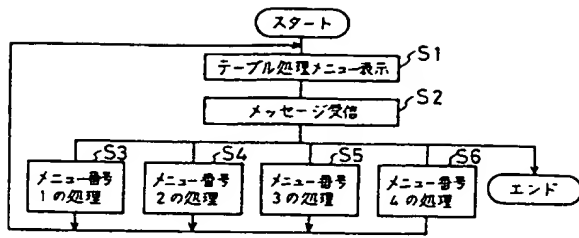
本発明の実施例のブロック図

第1図



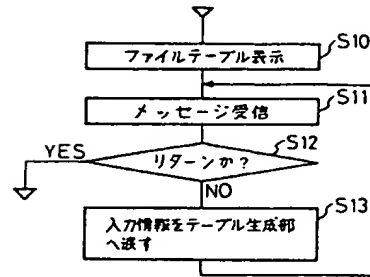
ファイル制御テーブル14の構成例を示す図

第2図



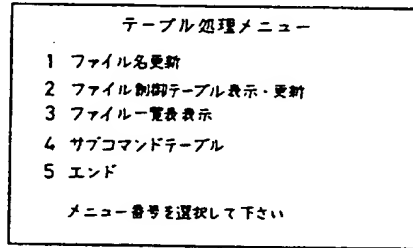
メニュー制御部3の処理例の流れ図

第3図



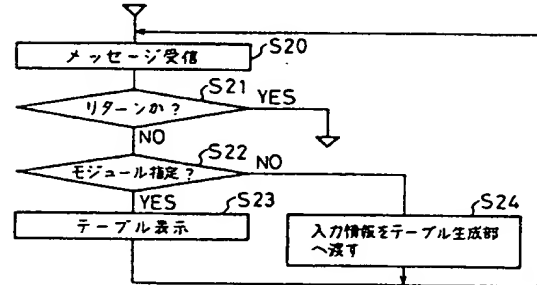
メニュー番号1の処理例の流れ図

第5図



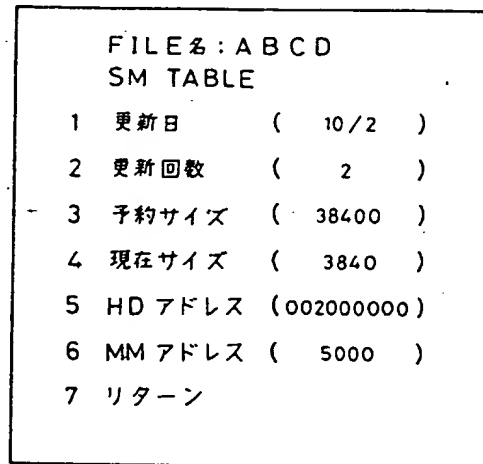
テーブル処理メニュー画面の内容例を示す図

第4図



メニュー番号2の処理例の流れ図

第6図



SM テーブル表示画面の内容例を示す図

第7図

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**